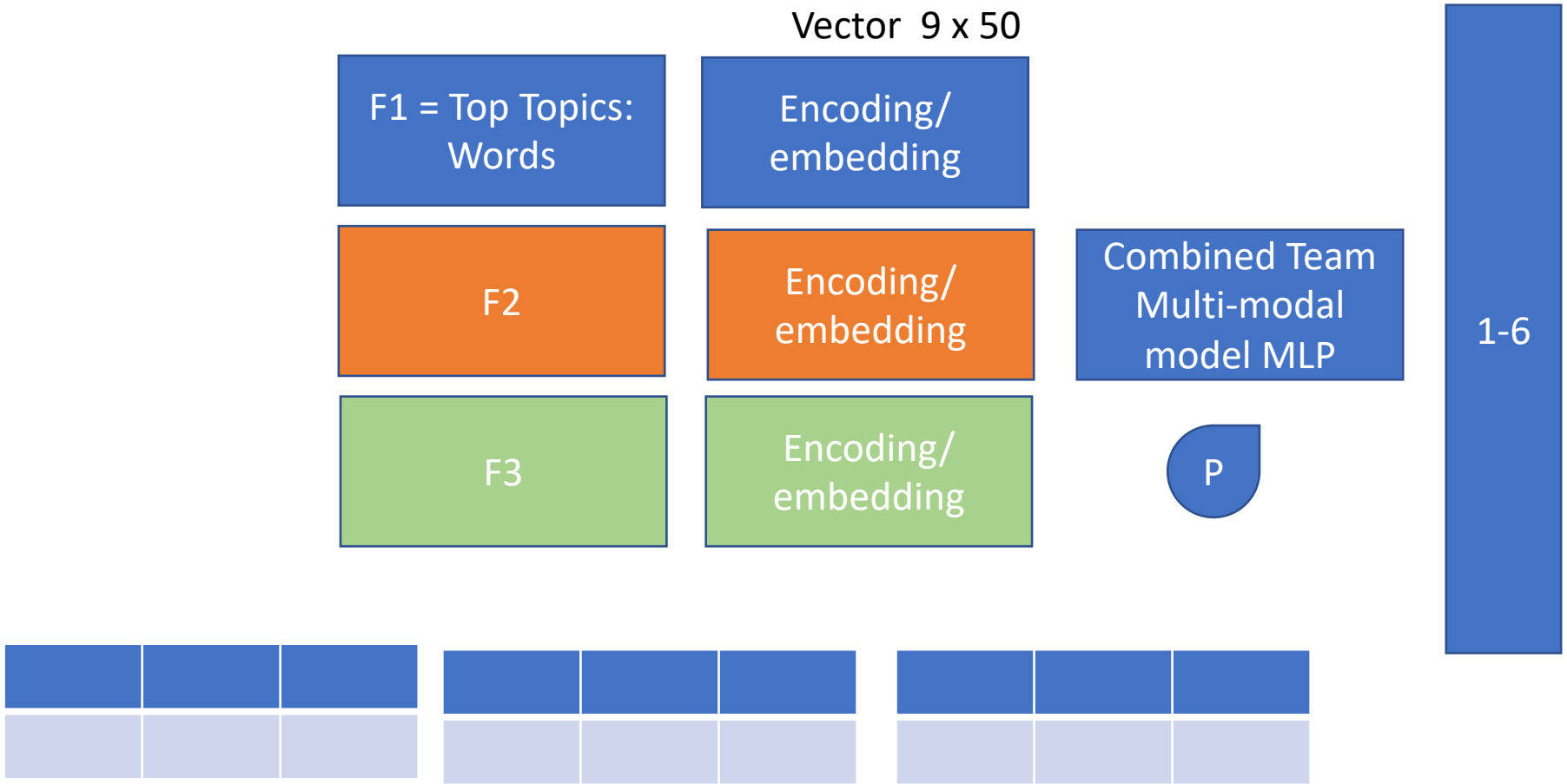
Week 14 – Insights Post Deployment and MLSE

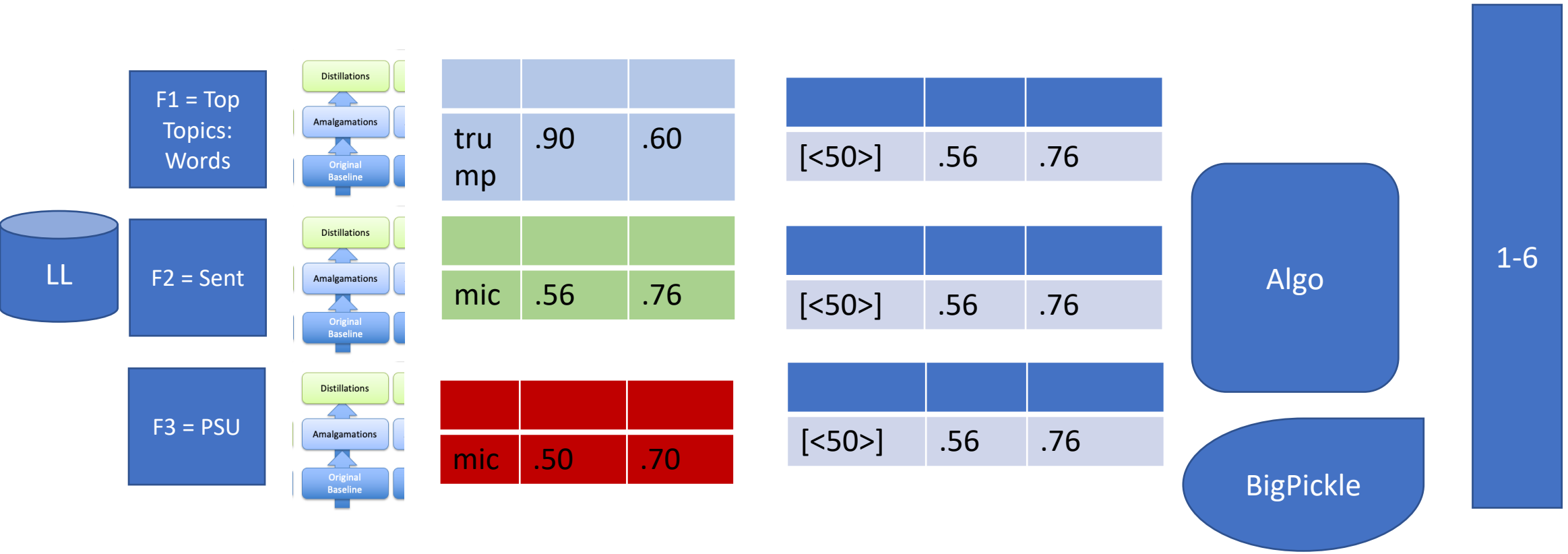
DrArsanjani

Lesson 14

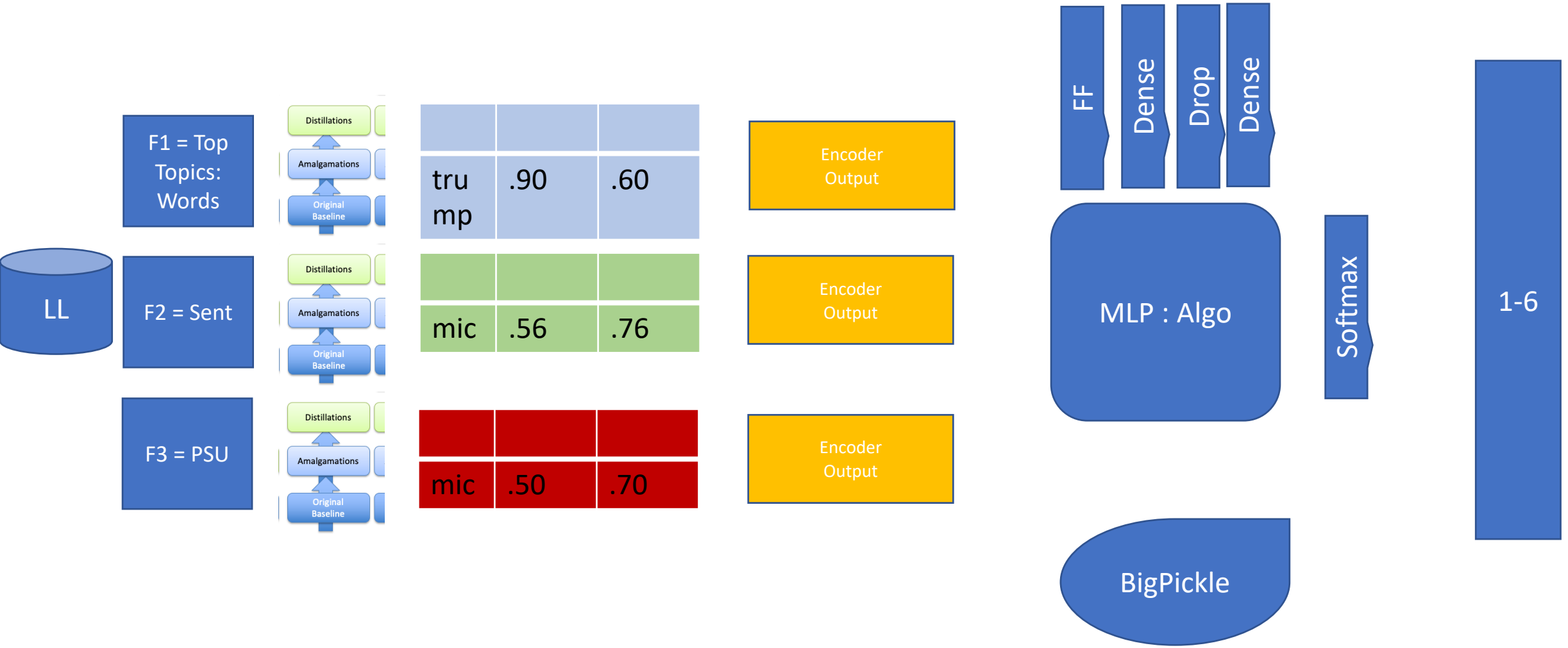
1. Post deployment : Scrape 50
    1. Inference not done when you train with a strong model
2. GUI like considerations?
    1. Here is a cnn.com website, watch I can scrape here in realtime
    2. Save to csv
    3. Now we can run inference
3. Output should be a 1-6; just like politifact!
    1. Challenges : each model may have diff output labels
4. You can change the weights based on reading the articles or headings and exercising human judgement
5. ML SE
    1. Import factor
    2. Pickle models
    3. Class
    4. [Git]
    5. Scrape from site to file , run inf from file
6. Explainability
    1. Factors, weights influence (Gini, feature_imp, Shapley values)


Vector 9 x 50
F1 = Top Topics: Words
Encoding/ embedding
F2
Encoding/ embedding
F3
Encoding/ embedding
Combined Team Multi-modal model MLP
P
1-6

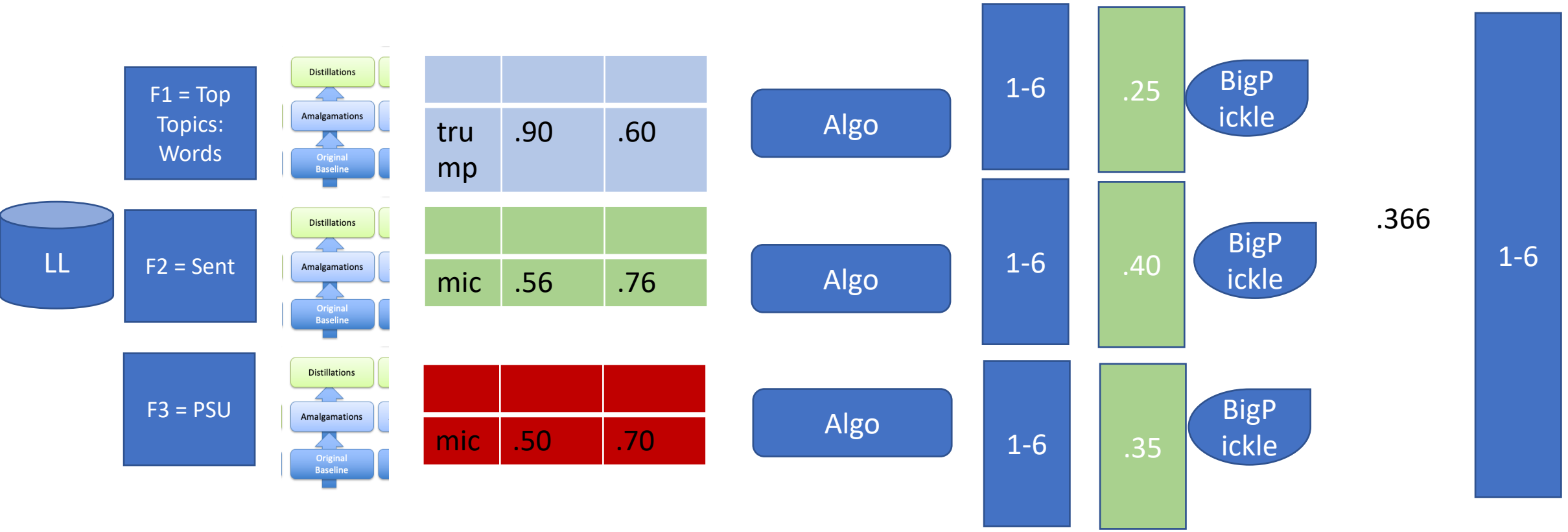
Training : EACH person DID THIS



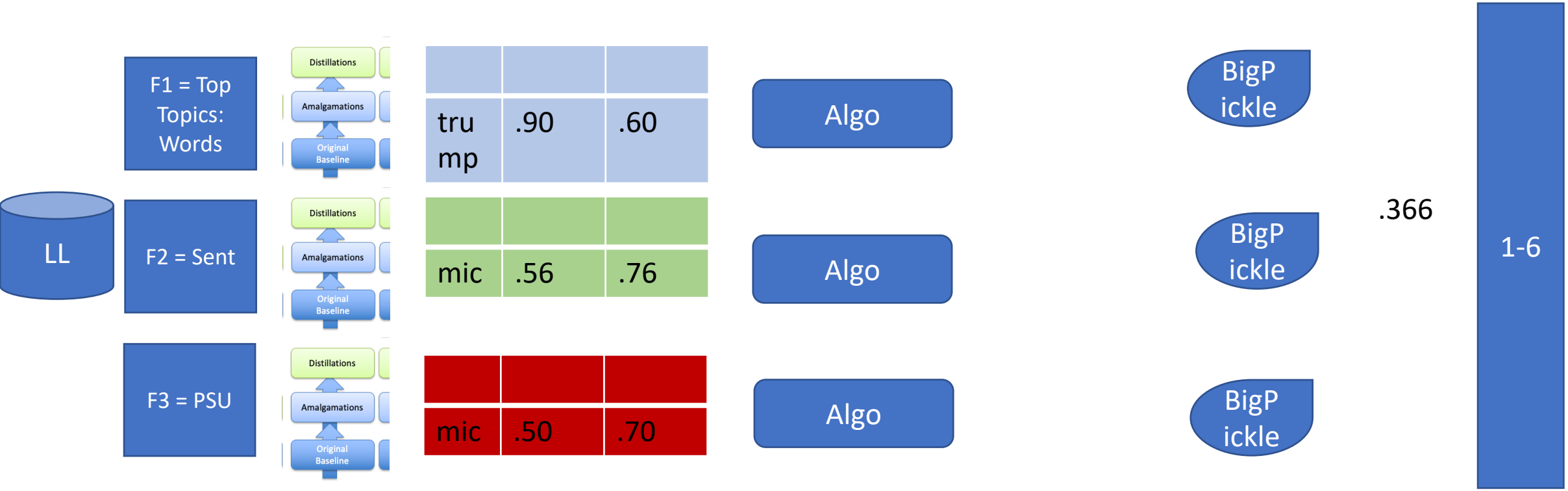
This is your assignment for next week

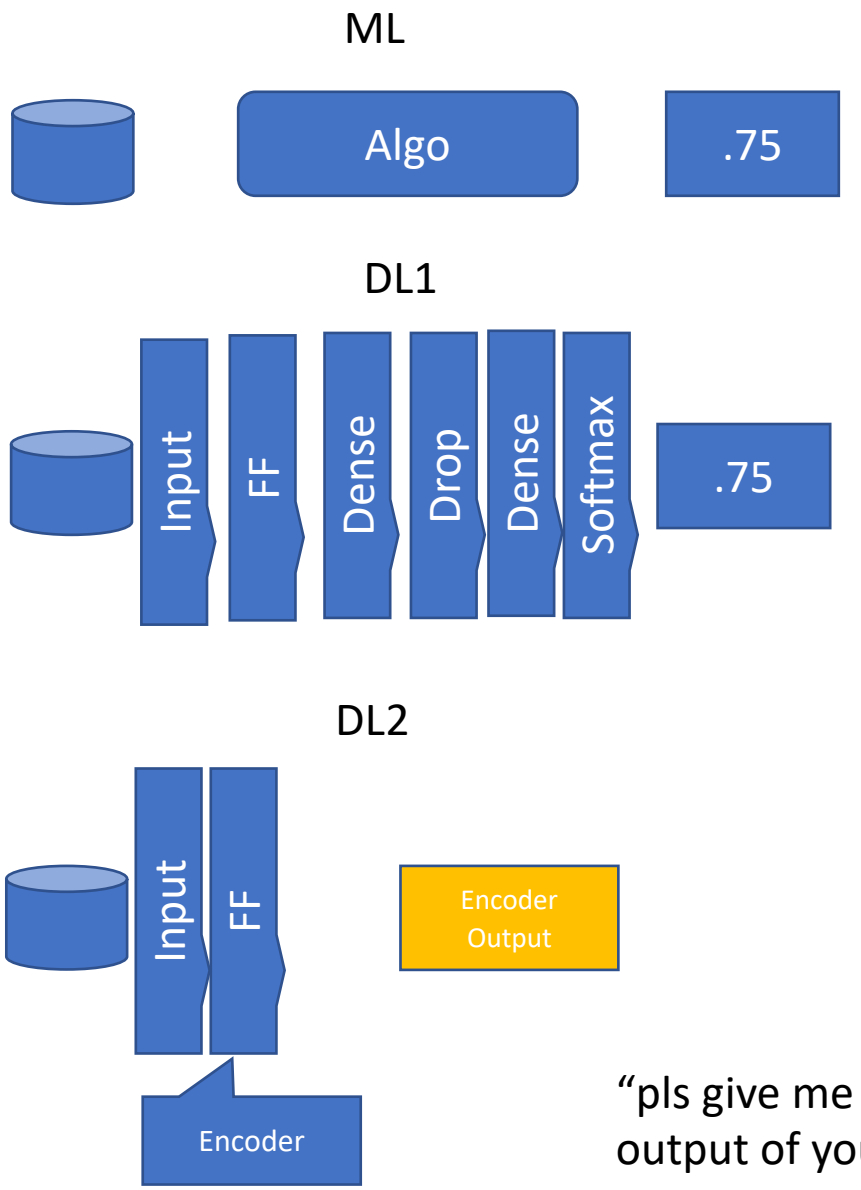


Training : EACH person DID THIS

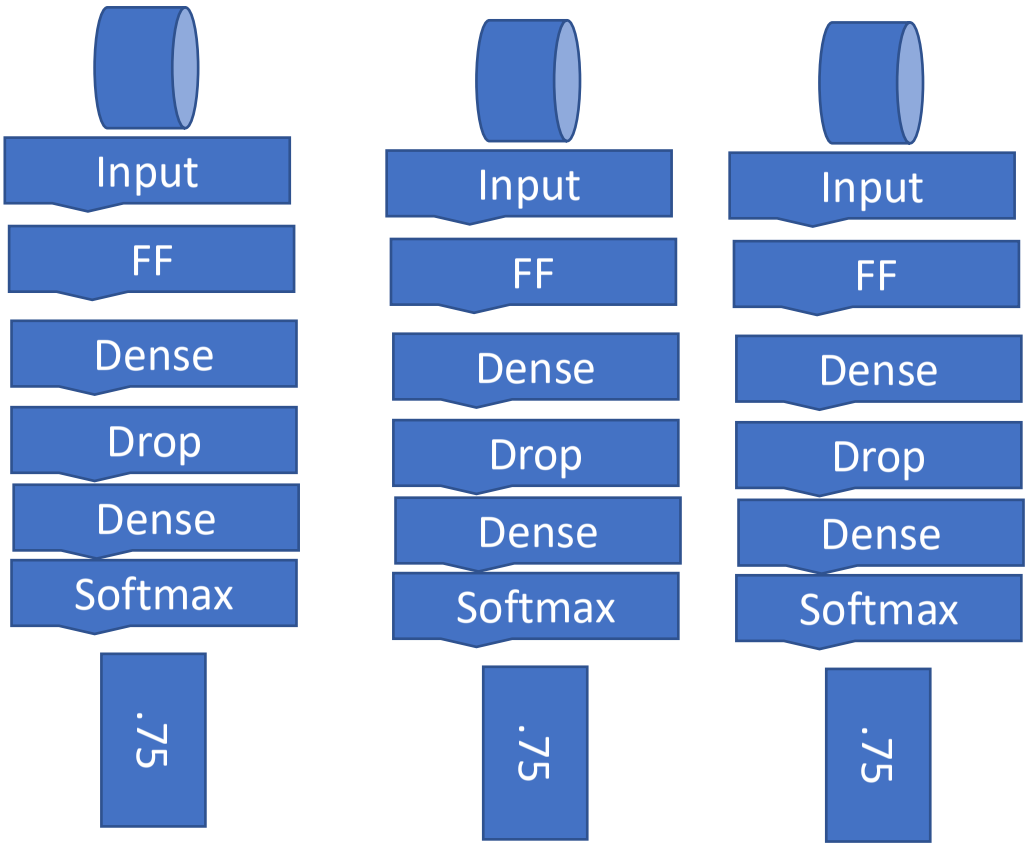


New Advanced Multi-modal

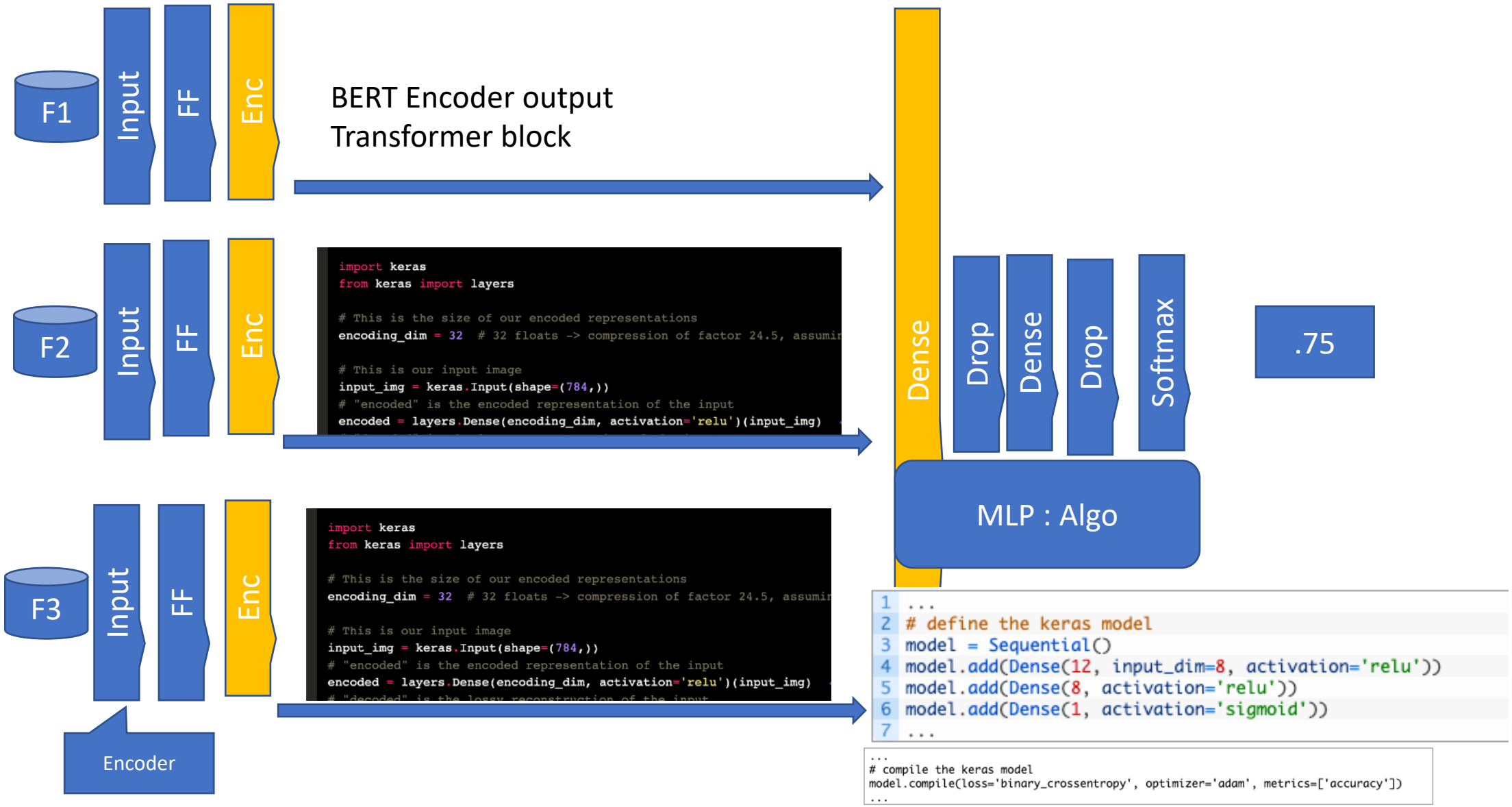




“pls give me the encoder output of your neural network”


Input
FF
Dense
Drop
Dense
Softmax
.75
Input
FF
Dense
Drop
Dense
Softmax
.75
Input
FF
Dense
Drop
Dense
Softmax
.75

```python
import keras
from keras import layers

# This is the size of our encoded representations
encoding_dim = 32  # 32 floats -> compression of factor 24.5, assumir

# This is our input image
input_img = keras.Input(shape=(784,))
# "encoded" is the encoded representation of the input
encoded = layers.Dense(encoding_dim, activation='relu')(input_img)
# "decoded" is the lossy reconstruction of the input
```



https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/

https://blog.keras.io/building-autoencoders-in-keras.html

In code:

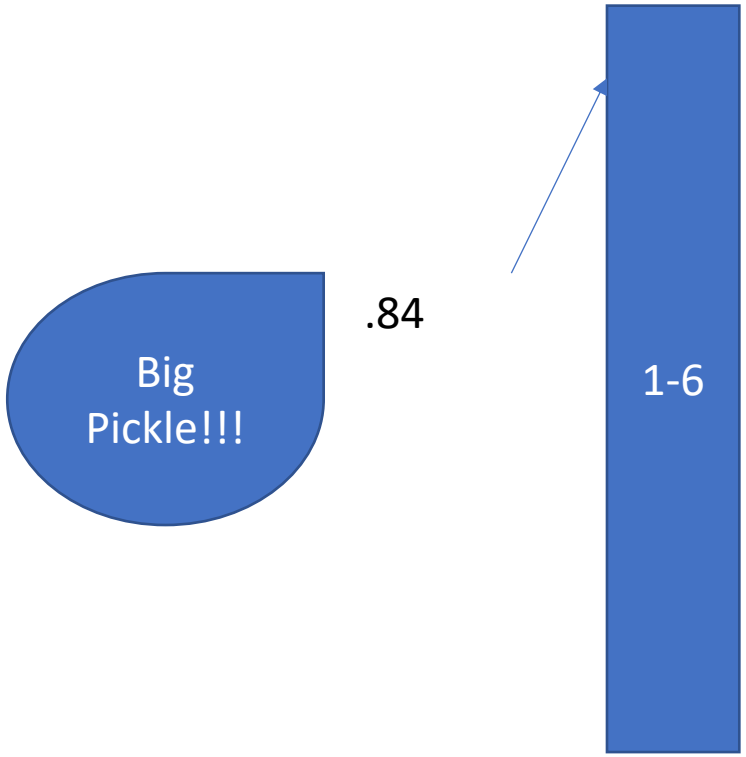
```
import keras
from keras import layers

# This is the size of our encoded representations
encoding_dim = 32  # 32 floats -> compression of factor 24.5, assuming the i

# This is our input image
input_img = keras.Input(shape=(784,))
# "encoded" is the encoded representation of the input
encoded = layers.Dense(encoding_dim, activation='relu')(input_img)
# "decoded" is the lossy reconstruction of the input
decoded = layers.Dense(784, activation='sigmoid')(encoded)

# This model maps an input to its reconstruction
autoencoder = keras.Model(input_img, decoded)
```

Let's also create a separate encoder model:

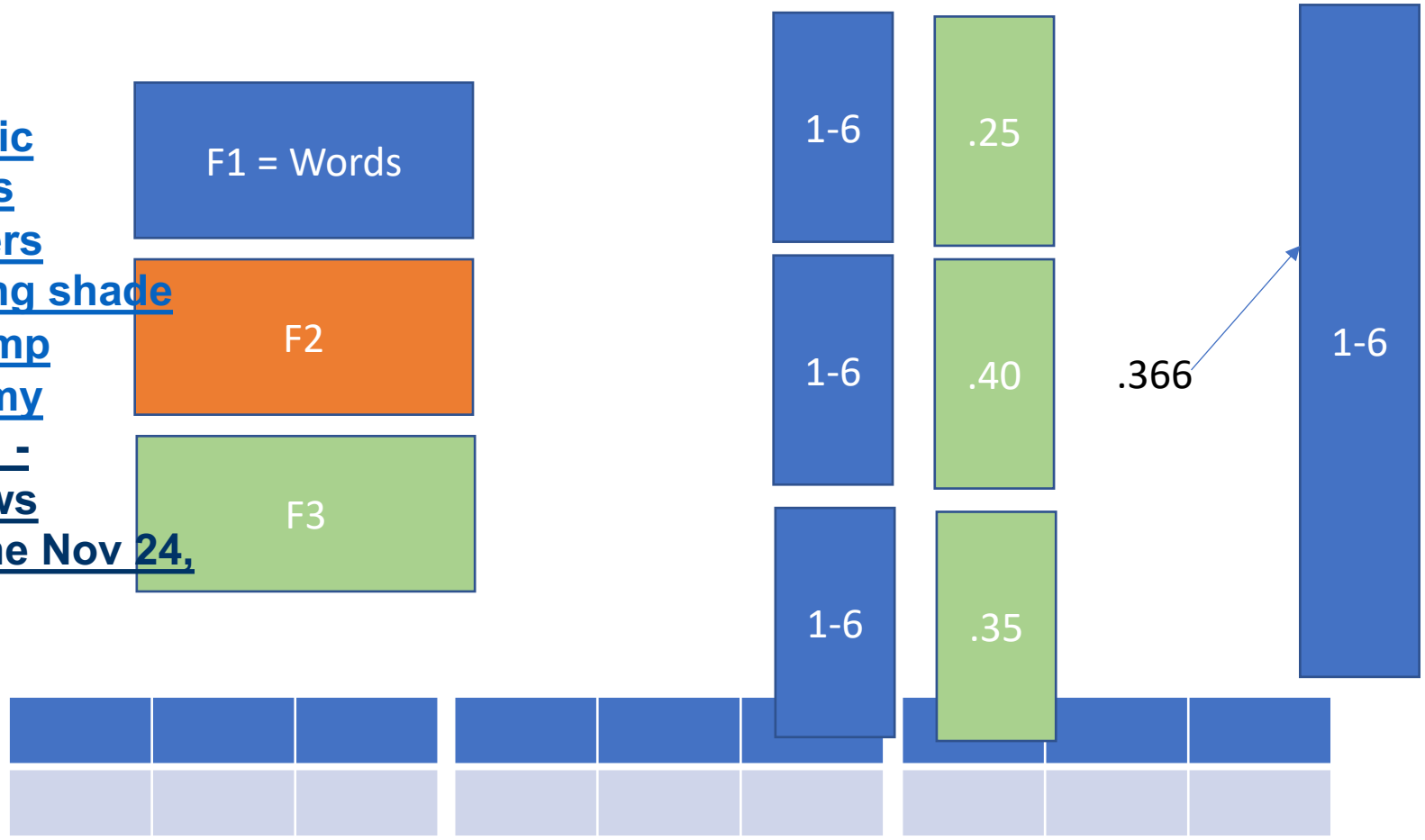
```
# This model maps an input to its encoded representation
encoder = keras.Model(input_img, encoded)
```

Inference

“Hot mic catches reporters throwing shade on Trump economy praise” - Foxnews headline Nov 24, 2020



“Hot mic catches reporters throwing shade on Trump economy praise” - Foxnews headline Nov 24, 2020

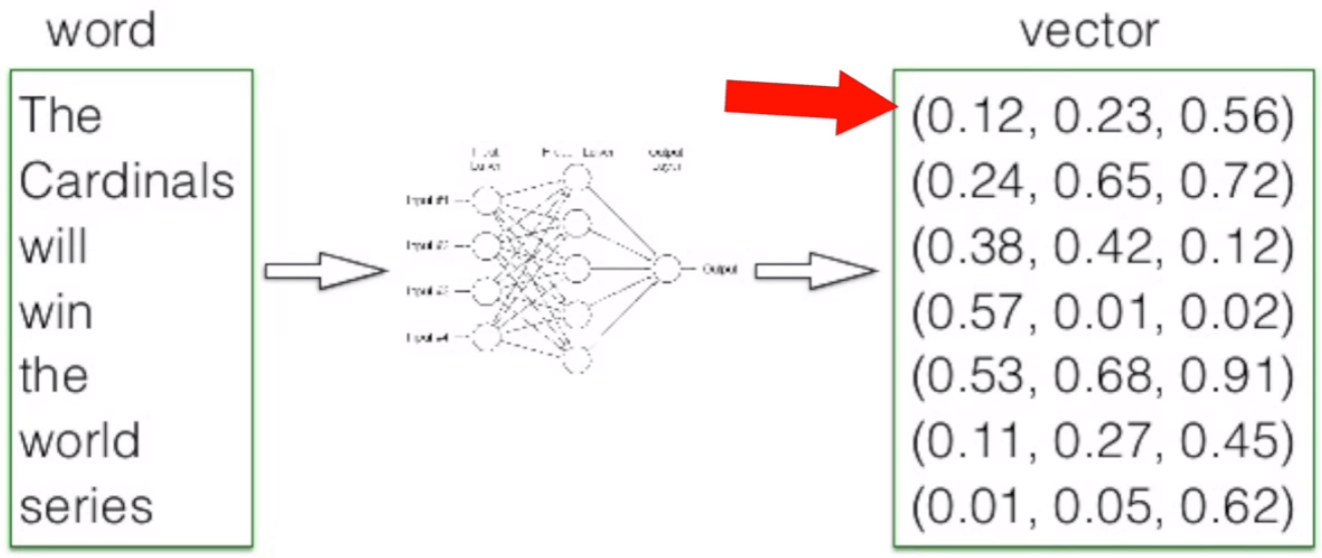


“Hot mic catches reporters throwing shade on Trump economy praise”

“this is the news headline” Vector 9 x 50

[‘this’,’news’, ‘headline’]

[Cove|FastText| BlazingText| Word2vec|Glove](‘this’,’news’, ‘headline’])



http://jalammar.github.io/illustrated-word2vec/

This is a word embedding for the word “king” (GloVe vector trained on Wikipedia):

```
[ 0.50451 , 0.68607 , -0.59517 , -0.022801, 0.60046 , -0.13498 , -0.08813 , 0.47377 , -0.61798 , -0.31012 ,
-0.076666, 1.493 , -0.034189, -0.98173 , 0.68229 , 0.81722 , -0.51874 , -0.31503 , -0.55809 , 0.66421 , 0.1961
, -0.13495 , -0.11476 , -0.30344 , 0.41177 , -2.223 , -1.0756 , -1.0783 , -0.34354 , 0.33505 , 1.9927 ,
-0.04234 , -0.64319 , 0.71125 , 0.49159 , 0.16754 , 0.34344 , -0.25663 , -0.8523 , 0.1661 , 0.40102 , 1.1685 ,
-1.0137 , -0.21585 , -0.15155 , 0.78321 , -0.91241 , -1.6106 , -0.64426 , -0.51042 ]
```

It’s a list of 50 numbers. We can’t tell much by looking at the values. But let’s visualize it a bit so we can compare it other word vectors. Let’s put all these numbers in one row:

Let’s color code the cells based on their values (red if they’re close to 2, white if they’re close to 0, blue if they’re close to -2):